

MacPherson Kwok Chen & Heid LLP

1762 Technology Drive, Suite 226

San Jose, CA 95110

Tel. (408) 392-9250

Fax (408) 392-9262

2402 Michelson Drive, Suite 210

Irvine, CA 92612

Tel. (949) 752-7040

Fax (949) 752-7049

Email: mailbox@macpherson-kwok.com

www.macpherson-kwok.com



August 12, 2005

Mail Stop Appeal Brief - Patents

Commissioner for Patents

P.O. Box 1450

Alexandria, VA 22313-1450

Re: Applicant: Foster, Robert A.
Assignee: Financial Systems Technology Pty. Ltd.
Title: Data Processing System For Pricing, Costing And Billing Of Financial Transactions
Application No.: 09/535,573 Filed: March 27, 2000
Examiner: C. Nguyen Group Art Unit: 3625
Docket No.: M-4540-1C US

Dear Sir:

Transmitted herewith are the following documents in the above-identified application:

- (1) Return Receipt Postcard;
- (2) This Transmittal Letter (in duplicate); and
- (3) Appeal Brief (pages 1-23) including Claims Appendix (pages 24-30) (30 pages).

☒ The fee has been calculated as shown below:

CLAIMS AS AMENDED

	Claims Remaining After Amendment		Highest No. Previously Paid For		Present Extra	Rate	Additional Fee
Total Claims	40	Minus	40	=	0	x \$50.00	\$ 0.00
Independent Claims	2	Minus	3	=	0	x \$200.00	\$ 0.00
<input type="checkbox"/> Fee of _____ for the first filing of one or more multiple dependent claims per application							\$ 0.00
<input checked="" type="checkbox"/> Fee for Filing a Brief in Support of an Appeal							\$ 500.00
<input type="checkbox"/> Fee for Request for Extension of Time							\$ 0.00
Total additional fee for this Amendment:							\$ 0.00
<input checked="" type="checkbox"/> Conditional Petition for Extension of Time: If an extension of time is required for timely filing of the enclosed document(s) after all papers filed with this transmittal have been considered, an extension of time is hereby requested.							
<input checked="" type="checkbox"/> Please charge our Deposit Account No. 50-2257 in the amount of							\$ 500.00
<input checked="" type="checkbox"/> Also, charge any additional fees required and credit any overpayment to our Deposit Account No. 50-2257							
Total:							\$ 500.00

I hereby certify that this correspondence is being deposited with the United States Postal Service as First Class Mail in an envelope addressed to: Mail Stop Appeal Brief - Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on August 12, 2005.

Edward C. Kwok
Attorney for Applicant(s)

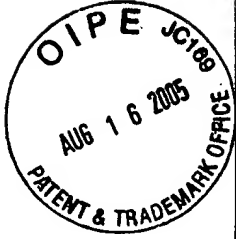
8/12/2005
Date of Signature

Respectfully submitted,

Edward C. Kwok
Edward C. Kwok
Attorney for Applicant
Reg. No. 33,938

Reg. No. 43,779

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE



Appellant: Robert A. Foster
Assignee: Financial Systems Technology Pty. Ltd.
Title: Data Processing System for Complex Pricing and Transactional Analysis
Serial No.: 09/535,573 Filing Date: March 27, 2000
Examiner: Cuong H. Nguyen Group Art Unit: 2165
Docket No.: M-4540-1C US

San Jose, California
August 12, 2005

Mail Stop Appeal Brief - Patents
Commissioner For Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF UNDER 37 CFR § 41.37

Dear Sir:

Appellant submits this Appeal Brief in support of the Notice of Appeal filed in this case on August 11, 2005.

LAW OFFICES OF
MACPHERSON KWOK CHEN &
JIEID LLP

1762 Technology Drive
Suite 226
SAN JOSE, CA 95110
TEL (408) 392-9250
FAX (408) 392-9262

I. REAL PARTY IN INTEREST

The real party in interest is Assignee Financial Systems Technology Pty. Ltd.

II. RELATED APPEALS AND INTERFERENCES

There are no appeals or interferences known to Appellant, Appellant's legal representative, or the Assignee which will directly affect or be directly affected by or have a bearing on the decision by the Board of Patent Appeals in the pending appeal.

III. STATUS OF CLAIMS

Claims 47-86 are pending, rejected and appealed.

IV. STATUS OF AMENDMENTS

The pending claims have been in substantially the current form since the time Appellant filed a Response to Final Office Action on February 6, 2003, in response to the Final Office Action of November 6, 2002 ("First Final Office Action"). Subsequently, the Examiner issued an Advisory Action on March 17, 2003, entering the amendment, but maintaining the Examiner's various rejections. In response, Appellant filed a Notice of Appeal on April 18, 2003 and an Appeal Brief on July 10, 2003. On November 4, 2003, the Examiner issued another Final Office Action ("Second Final Office Action"), which withdrew the First Final Office Action on the basis that "[the] Appeal Conference suggested that the previous Final Office Action was not presented in an organized manner..." In response,

Appellant filed a Supplementary Appeal Brief on March 2, 2004 and requested that the previous Appeal be Re-instated. The Examiner then issued a non-final Office Action on June 18, 2004, repeating substantially the same rejections as set forth in the First Final Office Action, and in addition, rejecting the claims on 35 U.S.C. § 101. Based on the 35 U.S.C. § 101 rejection, the Examiner denied Appellant's request to re-instate the previous Appeal. In response to this Office Action, Appellant filed an Amendment on October 18, 2004, amending the claims for formalities. The Examiner then issued another Final Office Action on April 11, 2005 ("Third Final Office Action"), repeating substantially all the rejections in the Office Action of June 18, 2004. Appellant filed a Notice of Appeal on August 11, 2005.

V. SUMMARY OF THE INVENTION

The present invention relates to an apparatus (e.g., independent Claim 68) and a method (independent Claim 47) for use in the financial services industry which provide a data processing system, including a database, suitable for pricing transactions. According to one embodiment, such as recited in Claim 47 and described in Applicant's Specification, beginning at page 7, line 21 to page 14, line 32, the method includes (a) creating a transaction instance corresponding to a transaction; (b) creating a first production service instance, linked by a first relation instance, representing an action performed to process the transaction; and (c) creating a billing service instance linked to the first production service instance by a second relation instance, representing a billing service related to a pricing of said first production service. Claim 68 recites corresponding means. In one embodiment, the method and means further include creating a second production service instance linked to the transaction instance by the first relation instance (see, for example, Claims 48, 50, 69 and 71

69 and Appellant's Specification, at page 9, lines 2-11). Multiple billing service instances can be linked to the production service instances by relation instances (see, for example, Claims 49-50 and 70-71, Appellant's Specification, beginning at page 12, line 9 to page 13, line 17).

The means and the method of the present invention allow effective transaction analysis. For example, the method allows creating relation instances linking a transaction instance to an account instance, a client instance, an entity instance, or a market segment instance (See, for example, Claims 52-55 and 73-76, Appellant's Specification, at page 16, lines 3-14).

In means and a method of the present invention, transaction instances, production service instances and billing service instances can be stored in one or more entity instance tables (See, for example, Claims 56 and 76 and Appellant's Specification, at page 10, lines 8-13). The relation instances can be stored in one or more relation instance tables (See, for example, Claims 57 and 77 and Appellant's Specification, at page 10, lines 13-15).

The method of the present invention can be enhanced to include linking of a settlement service instance to a billing service instance by a relation instance (See, for example, Claims 58 and 78 and Appellant's Specification, beginning at page 21, line 30 to page 23, line 10). Price tables instances, including cost and fee tables, can also be created and linked to transaction instances and billing instances (See, for example, Claims 59-67 and 79-86 and Appellant's Specification, beginning at page 23, line 11 to page 24, line 24).

The apparatus of the present invention is provided to carry out the methods of the present invention.

VI. GROUND OF REJECTION TO BE REVIEWED

1. Whether or not the Examiner erred in rejecting Claims 47-67 under 35 U.S.C. § 101 for being directed to non-statutory matter.
2. Whether or not the Examiner erred in rejecting Claims 47-54, 56-66, 68-86 under 35 U.S.C. § 103(a) over U.S. Patent 5,630,127 ("Moore"), in view of U.S. Patent 5,682,482 ("Burt").
3. Whether or not the Examiner erred in rejecting Claim 55 under 35 U.S.C. § 103(a) over Moore, in view of Burt, and further in view of U.S. Patent 5,636,117 ("Rothstein").
4. Whether or not the Examiner erred in rejecting Claims 64, 66 and 85 under 35 U.S.C. § 103(a) over Moore, in view of Burt, and further in view of U.S. Patent 5,559,313 ("Claus").

VII. ARGUMENTS

1. Whether or not the Examiner erred in rejecting Claims 47-67 under 35 U.S.C. § 101 for being directed to non-statutory matter.

In the Office Action of June 18, 2004, the Examiner rejected Claims 47-67 and 68-86 under 35 U.S.C. § 101 based on the Examiner's belief that these claims are directed to non-statutory matters. With respect to Claims 47-67, the Examiner states, in pertinent part, that "claim 47 is ONLY useful when a computer is not running a claimed method (in claim 47) can not be derived (i.e., merely claiming a floppy disk with claimed instructions) ..."

(emphasis in the original). With respect to Claim 68-86, the Examiner states, in pertinent part, that “this claim contains computer-per-se materials, although a processing system is claimed.”

In response, in the Amendment of October 18, 2004, Appellants amended Claims 47-52, 54, 58-65 and 67 to no longer recite a “computer readable medium” and Claim 68 to further recite “means for computing a price”. As amended, Claims 47 and 68 recite:

47. A method for providing a database suitable for pricing transactions, the method comprising:

creating a transaction instance corresponding to a transaction;

creating a first production service instance representing an action performed to process said transaction, said first production service instance being linked to said transaction instance by a first relation instance;

a billing service instance representing a billing service related to a pricing of said first production service, said billing service instance being linked to said first production service instance by a second relation instance; and

pricing said transaction for billing a customer based on said transaction instance, said first production service instance and said billing service instance.

68. A database data processing system for pricing transactions, said data processing system comprising:

means for creating a transaction instance corresponding to a transaction;

means for creating a first production service instance representing an action performed to process said transaction, said first production service instance being linked to said transaction instance by a first relation instance;

means for creating a billing service instance representing a billing service related to a pricing of said first production service, said billing service instance being linked to said first production service instance by a second relation instance; and

means for computing a price for said transaction, said price being a cost to a customer for said transaction computed based on said transaction instance, said first production service instance and said billing service instance.

Therefore, Claims 47 and 68 recite, respectively, a method and a database that price a financial transaction. As discussed in Appellant's Specification, beginning at page 1, line 30 to page 4, line 5, pricing and costing financial transactions are difficult tasks necessary for managing profitability in large financial institutions (e.g., commercial and investment banks). Profitability management has a great impact on these institutions, especially where sophisticated clients with numerous and complex financial transactions are involved. In a large commercial bank, hundreds of millions of transactions are routinely carried out in a given business day. The present invention, as set forth in Claims 47 and 68, allows and provides means for such transactions to be individually analyzed for cost and priced, which is a concrete, tangible and useful result.

According to the case law, set forth, for example, in the MPEP § 2106(IV)(B)(2)(b)(ii), a claimed process is statutory if it is limited to a practical application of an abstract idea or mathematical algorithm in the technological arts. In *State Street Bank & Trsut v. Signature Financial Group*, 149 F.3d at 1373, 47 USPQ2d at 1601, the Court of Appeals, Federal Circuit, teaches how to determine such a practical application:

A claim is limited to a practical application when the method, as claimed produces a concrete, tangible and useful

LAW OFFICES OF
MACPHERSON KWOK CHEN &
HEID LLP

1762 Technology Drive
Suite 226
SAN JOSE, CA 95110
TEL (408) 392-9250
FAX (408) 392-9262

result; i.e., the method recites a step or act of producing something that is concrete, tangible and useful. See AT&T, 172 F.3d at 1358, 50 USPQ2d at 1452. Likewise, a machine claim is statutory when the machine, as claimed, produces a concrete, tangible and useful result (as in *State Street*, 149 F.3d at 1373, 47 USPQ2d at 1601) ...

Thus, Appellant submits that Claims 47 and 68, and their respective dependent Claims 48-67 and 69-86 are each statutory subject matter that is unambiguously within the technological art of pricing a transaction, as a concrete, tangible and useful result – i.e., the result of a customer being fairly billed for the service or services rendered in each transaction – is achieved.

In response to Appellant's arguments, in paragraphs 11 and 12 of the Third Final Office Action, the Examiner maintains his rejections of Claims 47-57 under 35 U.S.C. § 101. (The Examiner apparently withdrew his rejection of Claims 68-86 under 35 U.S.C. § 101.) With respect to Appellant's arguments, the Examiner responds:

In the previous Office Action, the examiner states "claim 47 may not be concrete and tangible when a computer is not running" (i.e., claimed steps do not exist if no execution), he means that the pending claimed invention MUST require an operational computer for performing those steps. The 35 USC 101 panel advised on Wed., 1/12/2005 that the amended claim 47 is still not statutory; a requirement for technological art on 35 USC 101 (previous rejections were not included because "technological art" not enforced).

Appellant respectfully submit that the Examiner's response is simply unintelligible. When the computer is not running, the claimed methods in Claims 47-67 are not carried out. Appellant believes that the test stated in MPEP § 2106(IV)(B)(2)(b)(ii) requires a determination of whether or not the claim is limited to a practical application when the

claimed invention is carried out. There is no point in determining whether a process claim is statutory when it is not carried out.

Therefore, Appellant respectfully requests that the Board of Appeal to reverse the Examiner's rejection of Claims 47-67 under 35 U.S.C. § 101.

2. Whether or not the Examiner erred in rejecting Claims 47-54, 56-66, 68-86 under 35 U.S.C. § 103(a) over U.S. Patent 5,630,127 ("Moore"), in view of U.S. Patent 5,682,482 ("Burt").

In the Office Action of June 18, 2004, in numerous lengthy discussions set forth in the paragraphs 7-18 and 21-32, the Examiner rejected Claims 47-54, 56-66 and 68-86 under 35 U.S.C. 103(a) over Moore and Burt. In rejecting each claim, the Examiner relied on his following analysis of Moore and Burt, which he substantially maintained since before the First Final Office Action of November 6, 2002 to the Third Final Office Action:

Here each instance having data identifying particular items such as transaction, production service, billing service. These data qualifies as non-functional descriptive material.

These claimed descriptive material are not functionally related to a substrate (e.g., a computer-readable medium). Rather those are just "being held" in the medium. As a result, those data can be called non-functional descriptive material and does not limit the claims.

Moore et al. ('127) disclose that a rule-based application structure could be a relational database where records of a transaction are related/linked to each other (see Moore, the abstract, Figs.3,4). Moore et al. teach that: service instances linking to transaction instances; creating a billing service instance linked to a service instance with relation instance (see Moore, "FIG. 4 is an object instance table." 6:54-59 "An example of this table is shown in FIG. 3. The names or "objects" are shown in columns "OBJECT" 302, "OBJECT1" 304 and "OBJECT2" 308. These names or "objects" stand for a

multitude of particular instances of the data, any of which can be retrieved by specifying the identifiers of the entities listed above which would focus the access on a particular representation value.” 10:5-19; 10:45-55 “An additional feature of the GRMS architecture is the placement of GRMS processor on the Business Professional’s workstation 118 along with the Object Table 300, and the program defined in the object table 300. Since the object instance table 400 is also present, the Business Professional can change values in the Object instance table (via GRMS screens and functions) and reprocess the report on the workstation. All object accesses will be satisfied by the Object instance table function and therefore, the CMIM database 224 is not needed for this “What if” analysis reporting”; in OOP, “instance” is a variable name e.g., service instance, relation instance etc.).

Moore et al. teach about a financial institution, and a single transaction can generate many object instances (see Moore et al., 1:21-30 and Detailed Description Text portion (para. 439) “Unique identifier for a GRMS transaction. A single GRMS transaction can generate many object instances”), Moore et al. do not explicitly disclose that financial transaction functions are connected together.

Burt et al. further disclose a system with related functions including financial transaction functions connected together (e.g. see Burt et al., Fig. 5, the abstract, 4:25-27, and 25:2-16), comprising:

- creating a transaction instance corresponding to a financial transaction (e.g. see Burt et al., Fig. 5, the abstract, col. 6 lines 1-14, and col. 21 lines 42-59)

The examiner submits that because Moore et al. teach applications using OOP macros wherein “instance” is a variable instance – an instance is a single occurrence of a class -, it would be obvious for the analogous use of macros: “transaction instance”, “service instance”, and “billing service instance”. Artisan would recognize that a total price for a transaction, including any changes of those instance variables (i.e., pricing a transaction “based on”: a transaction instance, production service instance, and billing service instance).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to create different instances as shown in Moore et al. and Burt et

al. because a total price for a transaction would take into account all predetermined pricing components related to said transaction (see in re Gulack, 703 F.2d 1381, 217 USPQ 401, 404 (Fed. Cir. 1983)).

First, Appellant disagrees with the Examiner that the “transaction instance”, “production service instance” and “billing service instance” are “non-functional descriptive material.” Appellant’s Specification, at page 9, lines 15-21, provides that a “transaction instance” is a representation of a transaction in the data processing system, a “production service instance” is the representation of a specific production service performed by the financial service company (FSC). At Appellant’s Specification, page 8, lines 9-27, Appellant explains that production services are “individual actions that the FSC must perform, or that the FSC wishes to count, to process the transaction completely,” and provides numerous examples of such production services. Similarly, at Appellant’s Specification, at page 12, lines 25-30, Appellant provides that a “billing services” as “activities which involve costs to the FSC or activities which involve fees to be charged.” Thus, the “billing service instance” is a representation of such activities in the data processing system. Thus, Appellant’s Specification makes clear how these terms should be construed and understood. However, not once does the Examiner refer to Appellant’s Specification to construe these terms. This is impermissible. As explained in *Phillips v. AWH Corporation*, Case 03-1269, -1286, (Fed. Cir. 2005), the Examiner must construe these claim terms in view of the Specification:

The claims, of course, do not stand alone. Rather, they are part of “a fully integrated written instrument,” Markman, 52 F.3d at 978, consisting principally of a specification that concludes with the claims. For that reason, claims “must be read in view of the specification, of which they are a part.” Id. at 979. As we stated in Vitronics, the specification “is always highly relevant to the claim construction analysis. Usually, it is

dispositive; it is the single best guide to the meaning of a disputed term.” 90 F.3d at 1582.

Thus, the Examiner’s contention that the term “transaction instance”, “production service instance” and “billing service instance” are non-functional descriptive material has no basis in fact or law.

At the outset, in a response filed on February 6, 2003, Appellant explained to the Examiner that the Examiner’s contentions regarding Moore’s teachings are unsupported. This is because the Examiner failed to show where in Moore’s disclosure is it disclosed or suggested the “service instances,” or “billing service instance.” In fact, Appellants submitted to the Examiner that Moore does not disclose “service instances” or “billing service instances.” Specifically, Appellant’s illustrated this failure to disclose “service instances” and “billing service instances” by referring to Moore’s specification, at col. 3, lines 40-59, where Moore merely discloses that its GRMS system is a risk management system that requires such data as foreign exchange rates, market prices and counter party ratings. Appellant explained that such information is qualitatively different from and provides no teachings relevant to the “production service instance” and the “billing service instance” recited in Claim 47, which are specific types of instances relating to pricing of a financial transaction:

Creating a transaction instance corresponding to a transaction;

Creating a first production service instance representing an action performed to process said transaction, said first production service instance being linked to said transaction instance by a first relation instance; and

Creating a billing service instance representing a billing service related to a pricing of said first production service, said

billing service instance being linked to said first production service instance by a second relation instance.

Yet, the Examiner still insisted, based on his view that Moore teaches “applications using OOP macros wherein ‘instance’ is a variable instance – an instance is a single occurrence of a class,” as quoted above, that “it would be obvious for the analogous use of macros: ‘transaction instance’, ‘service instance’ and ‘billing service instance’.” What suggestion in the prior art allows the Examiner to make that inference, the Examiner did not say. In fact, not only is there no such suggestion, the portions of Moore that the Examiner relied for his rejection, i.e., Figs. 3 and 4, the abstract and cols. 4-10, teach “option value,” “option exposure” and other data structures relating to currency exchange transactions in a rule-based system for currency trading using a relational database, which are entirely irrelevant to the subject matters of Claim 47. Moore neither discloses nor suggests any of transaction instances, production service instances, and billing service instance. Thus, Moore provides no teaching with respect to Claim 47, as set forth above.

Burt also neither discloses nor suggests the above-quoted limitations of Claim 47. In the portions of Burt that the Examiner relied on for his rejection (i.e., Fig. 5, the abstract, cols. 4, 6, 21 and 24-25), Burt merely discloses “a network 10 is provided that includes a number of support system 14.” (col. 4, lines 42-43). Figure 5 “summarizes certain that is required by the agents of the four network layers, as applied to billing functions.” (col 24, lines 56-60). The functions are managed by management, fulfillment, charging and booking “agents.” (col. 21, lines 42-59; col. 24, line 60 to col. 25, line 16). Thus, the combined teachings of Moore and Burt do not disclose or suggest Appellant’s Claim 47.

For substantially the same reasons as stated with respect to Claim 47, the combined teachings of Moore and Burt also neither disclose nor suggest Claim 68.

Appellant submits that the Examiner cobbled Moore and Burt together, even though the references pertain to different subject matters and having no relevant teaching with respect to each other or with respect to the subject matters of Claims 47-86 (i.e., database for pricing transactions). As a result, no coherent teaching can be synthesized from this collection of references, taken as a whole. Further, the Examiner's cited motivation for combining the teachings of Moore and Burt is: "a total price for a transaction would take into account all predetermined pricing components related to said transaction." That teaching is found in neither Moore nor Burt. Thus, there is no motivation or suggestion in these references to combine their teachings in the manner suggested by the Examiner. In fact, the subject matters cannot be so combined simply because neither Moore nor Burt teaches the instances the Examiner contends that they teach. Accordingly, Appellant submits to the Examiner that Claims 47-86 are each allowable over the references of record, whether considered individually or in combination.

With respect to the Examiner's rejections of Claims 48-54, 56-66, 69-75 and 77-85. The Examiner's reasons are also inadequate. In addition to the limitations of their parent Claim 47, which is allowable over Moore and Burt, as explained above, Claims 48-54 and 56-66 recite "second production service instance", "second relation instance", "third relation instance", "second billing service instance", "account instance", "fourth relation instance", "entity instance table", "relation instance table", "settlement service instance", "price table instance", "cost table instance", "fee table instance", and "mandatory relation instance", none

of which are disclosed or suggested by Moore or Burt. As explained in Applicant's Specification, at page 25, lines 3-12, for example, using the recited structures allows pricing of complex financial transactions. Neither these structures nor their attendant benefits are disclosed nor suggested by Moore or Bert. Accordingly, Applicant respectfully submits that Claims 48-54 and 56-66 are each allowable over the teachings of Moore and Bert, whether considered individually, or in any combination. For substantially the same reasons, Claims 69-75 and 77-85 are allowable over Moore and Burt, whether considered individually and in combination.

In the Third Final Office Action, in response to Appellant's arguments presented in the Amendment of October 19, 2004, the Examiner states:

Since there is no explanation in the claim of what an instance besides naming different instances, this concept was read-on by cited references (the examiner respectfully requests an amended claim to show how to create, and how to price a transaction other than claiming "broad" steps of creating instances); applicant argues on page 19 that "The combination of general notions of financial transactions and specific programming techniques simply does not disclose or suggest specific data structures necessary for pricing complex transaction"; the examiner respectfully submits that he does not see any distinguished meanings about above "specific data structure" in claim 47 except calling different names/instances (please note also that claim 47 is a method claim - not a system that requires "specific data structure"). ... The applicant confirms that "instance" in this invention has a broad, normal meanings (see "REMARKS" submitted on 10/22/2004); therefore, the cited references already comprise that "broad, and normal" meanings because this is a specific application of the term "instance" with its actual meanings. On page 19, the applicant confirms "None of Applicant's claims is limited by any structure of any programming language, object-oriented or otherwise;" ...

In the response received on 10/22/2004, the applicant

confirmed that the claimed term "instance" has no special meanings besides its original dictionary definition. Therefore, a broader, interpretation of claim 47 is established (previous interpretation - Office Action issued on 6/18/2004 - requires a "narrower" application of computer OOP for this special reserved word "instance").

The Examiner is incorrect that Applicant's claims do not recite specific data structures. As discussed above, the Specification provides that a "transaction instance" is a representation of a transaction in the data processing system, a "production service instance" is the representation of a specific production service performed by a FSC, and a "billing service instance" is a representation of such activities in the data processing system. One skilled in the art would should read "representation ... in a data processing system" to be a data structure.

The Examiner continues,

5. The term of "being linked" is taught by Burt et al. as using "connection layer" and "connection instance" (see Burt et al., Fig.5). The examiner submits that Burt et al. teach a relational model as claimed (see Burt et al., Fig.5); "relational model is a data model in which the data is organized in relations (tables). This is the model implemented in most modern database management system"; therefore, banking transactions and related pricing were known to implement this model for the pending claimed relational structures (such use of a relational database management software have been applied for banking transactions, see Moore et al., the abstract, Figs.3,4, 10:5-34, 23:27-61 e.g., these para. indicate relationships of an object and access type with the value in the object instance entity).

Appellant respectfully submits that the Examiner's arguments regarding Burt and Moore using the relational data model have no relevance to Claims 47-86. The fact remains that neither Moore nor Burt teach the "transaction instance", "production service instance" or "billing service instance" recited in Claims 47-86 that allow accurate pricing of production

services that allows an FSC to manage profitability. And, the Examiner is well-aware of these deficiencies in the cited references Moore and Burt. However, as is apparent from the Examiner's paragraphs 8 and 9 of the Third Final Office Action, the Examiner simply dismisses Appellant's contentions on an unintelligible basis. In the Examiner's mind, it appears, because the term "instance" is used in some object-oriented programs, all application of instances are obvious:

7. The examiner respectfully submits that claims 47, and 68 comprise elements of a relational database structure, would utilize an instance variable in its object-oriented program (i.e., an instance is an instantiated object of a particular class), an object is something that can have properties and relations).

8. At the end, on pg. 5, 1st para., the applicant argues that cited references do not teach a "client instance" and a "market segment instance."

The examiner submits that one with ordinary skill in the art would understand that in an object-oriented program (e.g., Java, Visual Basic, C++ .etc.):

- an (entity) instance could be a client instance; an entity instance could be a market segment instance because in OOP, "instance" is a variable: an instance is an instantiated object of a particular class.

The examiner submits that cited prior art's limitations are not necessary spelled-out exactly claimed languages; analogous interpretations based on definition for functions of those terms .show that such claimed languages would be obvious for meaningful modifications in OOP using in cited art's situations.

9. All claimed limitations have been known since events for pricing transactions always "link" to related objects in a relational database. As the examiner presents that the claimed subject matter is obvious with one of skills in the art, different "instances" in above claims may be defined according to the use of a particular "instance" in an object-oriented program, in relation to the "class" to which it belongs; in other words,

instance variable is just a variable associated with an event/action/instance of a class in OOPs (a class is a template for a group of objects an object such as: client, market segment with similar behaviour, and a common inheritance).

The Examiner's position is irrational and simply has no basis in fact or law. It is a canon of claim interpretation that every word in the claim must be construed to be significant and not superfluous. In this rejection, the Examiner not only ignored limitations in Appellant's claims, he also ignored the deficiencies in the cited references. The difference between the claimed subject matter and the cited prior art reference is therefore so starkly great that no reasonable person of ordinary skill in the art would find obvious.

The Examiner also provides a lengthy discussion regarding programming techniques that is irrelevant to the claimed subject matter, but based on which he concludes that the invention has no inventive concept:

6. The Microsoft Computer Dictionary (published in 1996) defines a standardized meaning of a database wherein data components are linked together within that database as followings: linked list: In programming, a list of elements of a data structure connected by pointers. A singly linked list has one pointer in doubly linked list has two pointers in each node pointing to the next and previous nodes. In a circular list, the first and last nodes of the list are linked together; and link: To produce an executable program from compiled modules (programs, routines, or libraries) by merging the object code (assembly language object code, executable machine code) of the program and resolving interconnecting references (such as a library routine called by a program), or to connect two elements in a data structure by using index variables (index: A listing of keywords and associated data that point to the location of more comprehensive information, such as files and records on a disk/record keys in a database), or pointer variables (pointer: In programming and information processing, a variable that contains the memory location (address) of some data rather than data itself). The act of linking data/items from different parts in a database is in cited references of Moore et al., Burt et al.,

Rothstein, Clause et al., and they are a fundamental knowledge in database structure of OOPS; from that available computer programming knowledge the applicant uses it to apply for a specific use (i.e. for pricing transactions). Therefore, the invention does not teach any new inventive concept according to cited references.

The Examiner's contentions are irrelevant because Appellant's claims do not recite any programming techniques of the types the Examiner discussed above. The Examiner appears to be fixated on the concepts of macros, classes, instances in object-oriented programming language used in some relational database systems. Notwithstanding that such programming concepts are neither recited nor inherent in Applicant's claims, the Examiner persists in alleging, but fails to demonstrate how general notions of financial transactions and the concepts of object-oriented programming language render obvious Applicant's claims, which recite specific instances created in a database specific to solving the problem of pricing transactions involving complex components.

The Examiner's allegations are misplaced. None of Applicant's claims is limited by any structure of any programming language, object-oriented or otherwise. The combination of general notions of financial transactions and specific programming techniques simply does not disclose or suggest specific data structures necessary for pricing complex transactions. To show that Applicant's claims are obvious in light of the cited prior art references, the Examiner must show, from the teachings of the prior art itself, how the general notions of financial transactions of Moore and Burt and the programming language structures may be modified by one of ordinary skill in the art to obtain the specific data structures recited in Applicant's claims and how these specific claims can be utilized to price complex financial transactions. The Examiner simply fails to provide such a showing.

Accordingly, Appellant respectfully requests that the Board reverse the Examiner's rejection of Claims 47-54, 56-66 and 68-86 under 35 U.S.C. § 103(a) over Moore and Burt.

3. Whether or not the Examiner erred in rejecting Claim 55 under 35 U.S.C. § 103(a) over Moore, in view of Burt, and further in view of U.S. Patent 5,636,117 ("Rothstein").

The Examiner also rejected Claim 55 under 35 U.S.C. § 103(a) as being unpatentable over Moore, Burt and U.S. Patent 5,636,117 ("Rothstein"), the Examiner refers to his rejection of Claim 54 and further states:

Rothstein further teaches that a market segment instance could be an entity instance (see 2:8-10; 2:54-47; 3:9-12) (e.g., mortgage entities are linked to business models by indices in a program).

Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to combine specific applications of Moore et al., and Burt et al. in OOP financial transaction with Rothstein because they all suggest a systematic method that use "instance" in OOP to track components of costs and fees each time a financial transaction is processed. Artisan would recognize that an instant in OOP would be a variable to measure the impact of any changes from financial transactions by tracking those instance variables.

Appellant respectfully submit that the Examiner is incorrect. Claim 55 depends from Claims 47, 52 and 54 and thus is allowable for at least the reasons set forth above with respect to Claim 47. In addition, as recited in Claim 55, the market segment instance is linked to the transaction instance by a fourth relation instance, which is neither disclosed nor suggested by any of Moore, Burt and Rothstein.

The Examiner's reliance on Rothstein's col. 2, lines 8-10, lines 54-57 and col. 3, lines 9-12 to teach "market segment instance" is misplaced. As clearly set forth in Rothstein, at

col. 2, lines 1-3, Rothstein “provides a technique for monitoring the strength and trends of a real estate market, whether nationally or locally.” Thus, Rothstein’s disclosure has no bearing on the “market segment instance” of Claim 55, which relates to “a method for providing a database suitable for pricing transactions.”

Thus, Claim 55 is allowable over Moore, Burt and Rothstein, individually and in combination. Appellant respectfully requests the Board to reverse the Examiner’s rejection of Claim 55 under 35 U.S.C. § 103(a) over Moore, Burt and Rothstein.

4. Whether or not the Examiner erred in rejecting Claims 64, 66 and 85 under 35 U.S.C. § 103(a) over Moore, in view of Burt, and further in view of U.S. Patent 5,559,313 (“Claus”).

The Examiner rejected Claims 64, 66 and 85 under 35 U.S.C. § 103(a) as being unpatentable over Moore, Burt and U.S. Patent 5,559,313 (“Claus”), the Examiner refers to his rejection of Claim 84 and further states:

Claus et al. further express analogous instances in a database, the examiner submits that since they are considered as variable instances in OOPs (see Figs 6, 9-11, 13, 15) for analogous examples tat were claimed about:

- an entity instance could be an account instance;
- an entity instance could be a client instance;
- an entity instance could be a market segment instance.

Therefore it would have been obvious to one of ordinary skill in the art at the time of invention to combine certain applications to combine Moore et al., Burt et al., and Claus et al. in financial transaction with OO programming (or different applications using relational database) because they all suggest a systematic method that use “instance” in a structural database to track all the components of costs and fees each time a

financial transaction is processed. It has been recognized that a financial system would be able to measure profitability in a flexible manner and to measure the impact of any changes from banking clients by tracking those variables,

Appellant respectfully submits that the Examiner is incorrect. Claim 64 depends from Claims 47 and 63, Claim 66 depends from 47 and 65, and Claim 85 depends from 68 and 84, and thus each of Claims 64, 66 and 85 are allowable for at least the reasons set forth above with respect to Claim 47. In addition, as recited in Claims 64, 66 and 85, the account instance, the client instance, and the market segment instance, respectively, are each linked to the transaction instance by an entity instance, which is neither disclosed nor suggested by any of Moore, Burt and Claus.

The Examiner's reliance on Claus to teach "account instance", "client instance" and "market segment instance" is unsupported. Claus relates to "a smart card that is responsive to a list of items with individual prices that are received from a point of sale (POS) terminal during the individual transaction to automatically insert these items into expense categories." Thus, Claus too provides no teachings relevant to a database for pricing transaction, which is the subject matter of each of Claims 47-86.

Thus, Claims 64, 66 and 85 are each allowable over Moore, Burt and Claus, individually and in combination. In response to Appellant's arguments, the Examiner states in the Third Final Office Action:

The examiner also submits that suggestions for "Categorization of purchased items for each transaction by a smart card" had been discussed by Claus et al. in their patent (see Claus, claim 1, and 2:15 to 3:22), and a relational database of items/instances for different transactions were done in a financial software program.

Once again, the Examiner extracts irrelevant teachings from the cited reference. The fact that Claus relates to the relational databases and involves some kind of financial transaction is simply not instructive to Appellant's Claims 64, 66 and 85.

Appellant respectfully requests the Board to reverse the Examiner's rejection of Claims 64, 66 and 85 under 35 U.S.C. § 103(a) over Moore, Burt and Claus.

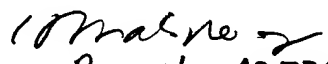
VIII. CONCLUSION

For the foregoing reasons, Appellant respectfully submits that Claims 47-86 are statutory and allowable the prior art of record. The Board of Patent Appeals and Interferences should therefore reverse the Examiner's rejections of these claims under 35 U.S.C. §§ 101 and 103(a).

I hereby certify that this correspondence is being deposited with the United States Postal Service as First Class Mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on August 12, 2005.


Attorney for Applicant(s)

8/12/2005
Date of Signature

Respectfully submitted,

Reg. No. 43,779
for Edward C. Kwok
Attorney for Applicant
Reg. No. 33,938

APPENDIX

Pending Claims 47-86 recite:

47. A method for providing a database suitable for pricing transactions, the method comprising:

Creating a transaction instance corresponding to a transaction;

creating a first production service instance representing an action performed to process said transaction, said first production service instance being linked to said transaction instance by a first relation instance;

creating a billing service instance representing a billing service related to a pricing of said first production service, said billing service instance being linked to said first production service instance by a second relation instance; and

pricing said transaction for billing a customer based on said transaction instance, said first production service instance and said billing service instance.

48. The method of claim 47, further comprising creating a second production service instance linked to said transaction instance by said first relation instance.

49. The method of claim 47, further comprising creating a second billing service instance linked to said first production service instance by said second relation instance.

50. The method of claim 47, further comprising creating a second production service instance linked to said transaction instance by a third relation instance.

51. The method of claim 47, further comprising creating a second billing service instance linked to said first production service instance by a third relation instance.

52. The method of claim 47, further comprising creating a third relation instance linking said transaction instance to an account instance.

53. The method of claim 52, wherein said account instance is linked to a client instance by a fourth relation instance.

54. The method of claim 52, further comprising creating a fourth relation instance linking said transaction instance to an entity instance.

55. The method of claim 54, wherein said entity instance is a market segment instance.

56. The method of claim 47, further comprising storing said transaction instance, said production service instance and said billing service instance in at least one entity instance table.

57. The method of claim 56, further comprising storing said first relation instance and said second relation instance in at least one relation instance table.

58. The method of claim 47, further comprising creating a settlement service instance linked to said billing service instance by a third relation instance.

59. The method of claim 47, further comprising:

creating a price table instance related to said transaction instance;

wherein said price table instance contains a price for said billing service instance.

60. The method of claim 59, wherein said price table instance is a cost table instance and said price is a cost.

61. The method of claim 59, wherein said price table instance is a fee table instance and said price is a fee.

62. The method of claim 61 further comprising creating a cost table instance related to said fee table instance by a mandatory relation instance.

63. The method of claim 47, further comprising:

creating an entity instance related to said transaction instance; and creating a price table instance related to said entity instance;

wherein said price table instance contains & price for said billing service instance.

64. The method of claim 63, wherein said entity instance is an account instance.

65. The method of claim 47, further comprising:

creating a first entity instance related to said transaction instance;

creating a second entity instance related to said first entity instance; and

creating a first price table instance related to said second entity instance;

wherein said first price table instance contains a price for said billing service instance.

66. The method of claim 65, wherein said first entity instance is an account instance and said second entity instance is a client instance.

67. The method of claim 65, further comprising creating a second price table instance related to first entity instance.

68. A database data processing system for pricing transactions, said data processing system comprising:

means for creating a transaction instance corresponding to a transaction;

means for creating a first production service instance representing an action performed to process said transaction, said first production service instance being linked to said transaction instance by a first relation instance;

means for creating a billing service instance representing a billing service related to a pricing of said first production service, said billing service instance being linked to said first production service instance by a second relation instance; and

means for computing a price for said transaction, said price being a cost to a customer for said transaction computed based on said transaction instance, said first production service instance and said billing service instance.

69. The data processing system of Claim 68, further comprising means for creating a second production service instance linked to said transaction instance by said first relation instance.

70. The data processing system of claim 68, further comprising means for creating a second billing service instance linked to said first production service instance by said second relation instance.

71. The data processing system of claim 68, further comprising means for creating a second production service instance linked to said transaction instance by a third relation instance.

72. The data processing system of claim 68, further comprising means for creating a second billing service instance linked to said first production service instance by a third relation instance.

73. The data processing system of claim 68, further comprising means for creating a third relation instance linking said transaction instance to an account instance.

74. The data processing system of claim 73, wherein said account instance is linked to a client instance by a fourth relation instance.

75. The data processing system of claim 68, further comprising means for creating a fourth relation instance linking said transaction instance to an entity instance.

76. The data processing system of claim 68, further comprising at least one entity instance table to store said transaction instance, said production service instance and said billing service instance.

77. The data processing system of claim 76, further comprising at least one relation instance table to store said first relation instance and said second relation instance.

78. The data processing system of claim 68, further comprising means for creating a settlement service instance linked to said billing service instance by a third relation instance.

79. The data processing system of claim 68, further comprising:

means for creating a price table instance related to said transaction instance;

wherein said price table instance contains a price for said billing service instance.

80. The data processing system of claim 79, wherein said price table instance is a cost table instance and said price is a cost.

81. The data processing system of claim 79, wherein said price table instance is a fee table instance and said price is a fee.

82. The data processing system of claim 81 further comprising means for creating a cost table instance related to said fee table instance by a mandatory relation instance.

83. The data processing system of claim 68, further comprising:

means for creating an entity instance related to said transaction instance; and
means for creating a price table instance related to said entity instance;

wherein said price table instance contains a price for said billing service
instance.

84. The data processing system of claim 68, further comprising:

means for creating a first entity instance related to said transaction instance;

means for creating a second entity instance related to said first entity instance;

and means for creating a first price table instance related to said second entity
instance;

wherein said first price table instance contains a price for said billing service
instance.

85. The data processing system of claim 84, wherein said first entity instance is an
account instance and said second entity instance is a client instance.

86. The data processing system of claim 84, further comprising means for creating
a second price table instance related to first entity instance.